Software for Arabic Machine Printed Optical Character Recognition (MACRS)

Mahdi. F. Al- Obaidi **Laheeb Mohammad Ibrahim**
Dept. of Math. *Dept. of Software*
College of Computer Sciences and Mathematics
University of Mosul, Iraq

Received on: 28/09/2002 **Accepted on: 04/01/2003**

ABSTRACT

Machine printed Arabic Character Recognition System (MACRS] is concerned with recognition of machine printed alphanumeric Arabic characters. In the present work, characters have been represented (extracted) by using geometric moment invariant (3 order). The technique used in this research can be divided into three major steps. The first step is digitization and preprocessing to create connected component, detect the skew of a character image and correct it. The second is feature extraction, where geometric moment invariant features of the input, Arabic character is used to extract features. Finally, we describe an advanced system of classification using probabilistic neural networks structure which yields significant speed improvements. MACRS is tested using 2961 patterns for a total 141 classes with roughly 21 patterns in each class. It is important to note here that the system performs extremely well with recognition rates ranging between 84% and 88% on different folds and the overall recognition is 85.8%. This is a very good performance taking into account the fact that we have a limited number of samples in each class and that, the recognition on the training data is also extremely high (99.8%) which represents a very good training.

Keyword: Machine printed Arabic Character Recognition System (MACRS] **,** probabilistic neural networks.

نظام تمييز الحروف العربيه المطبوعة بالآلة (MACRS)

مهدي فاضل موسئ **لهيب محمد ابراهيم**
كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ استلام البحث: 2002/09/28 **تاريخ قبول البحث: 2003/01/04**

الملخص

Machine printed Arabic Character Recognition System (MACRS) نظام لتمييز االحروف العربية المطبوعة، في هذا العمل تم استخلاص خواص الحروف العربية المطبوعة باستخدام طريقة العزوم **Geometric Moment Invariant (3 order)**. ان تقنية **MACRS** تُقَسَّمَ الى ثلاث خطوات رئيسيّة. الخطوة الأولى هي المعالجة

الاولية لتقسيم الوثيقة الصورية الى مقاطع صورية اصغر تمثل كل منها حرفاً ومن ثم تنحيف صورة الحرف وَاَكْتَشِافُ انحراف صورة الحرف و تُصَحِّحُه. الخطوة الثانية تمثل استخلاص الخواص حيث استخدمت طريقة العزوم في استخلاص الخواص للادخالات، واخيرا ، استخدم نظام متقدم للتمييز ويشمل استخدام الشبكات العصبيّة الاصطناعية (***Probablistic Neural Network)*** الّذي يَمْنَحُ سرعة مهمة في التمييز. اختبر **MACRS** على 2961 نَمُوذجاً تمثل 141 حرفاً عربياً تمثل الحروف ((ا –ي)، ء،لا، ة) بمواقع الحرف العربي الاربعة والارقام (0–9) فضلاً عن بعض الرموز (: ، . ، [،] ، ! ، = ، / ، * ، – ، + ، ؟ ، ،) بواقع 21 نَمُوذَجاً لكلّ حرف وقد لوحظ من خلال الاختبار أنّ النظام يَقُومُ بالتمييز بنسبة تَتَرَاوَحُ بين 84% و88% على مجموعات الاختبار المختلفة، عموما يتراوح المعدل العام للتمييز بنسبة 85.8% وهذا يعتبر أَدَاء جيّداً جدّا كذلك فان التمييز على معلومات التَدْرِيب تراوح مقداره 99.8% .

الكلمات المفتاحية: نظام لتمييز الحروف العربية المطبوعة، الشبكه العصبيه الاصطناعيه ***Probablistic***.

1- Introduction

The recognition of Arabic characters has been an area of great interest for many years, and a number of research papers and reports have already been published in this area. There are several major problems with Arabic character recognition: Arabic characters are distinct and ideographic, many structurally similar characters exist in the character set. Thus, classification criteria are difficult to generate, [Abuhaiba, 1994, Amin, 1997a, Amin, 1998, Amondon, 2000].

Arabic is a major world language spoken by 186 million people [Klassen, 2001]. Very little research has gone into character recognition in Arabic due to the difficulty of the task and lack of researchers interested in this field. As the Arab world becomes increasingly computerized and mobile, and technology becomes increasingly ubiquitous, the need for a natural interface becomes apparent. For the benefits of optical character recognition (OCR) and after careful study the problems for recognized Arabic characters, we construct an application (system) in this area that has received a good amount of attention to recognize machine printed Arabic character. The system *Machine Printed Arabic Character Recognition System (MACRS)* development in the device for Arabic character recognition to process many documents automatically. *MACRS* is developed for machine printed Arabic character recognition to process image documents according to the information socialization is in progress actively. Also the *moment invariants* method and neural networks (*probablistic*

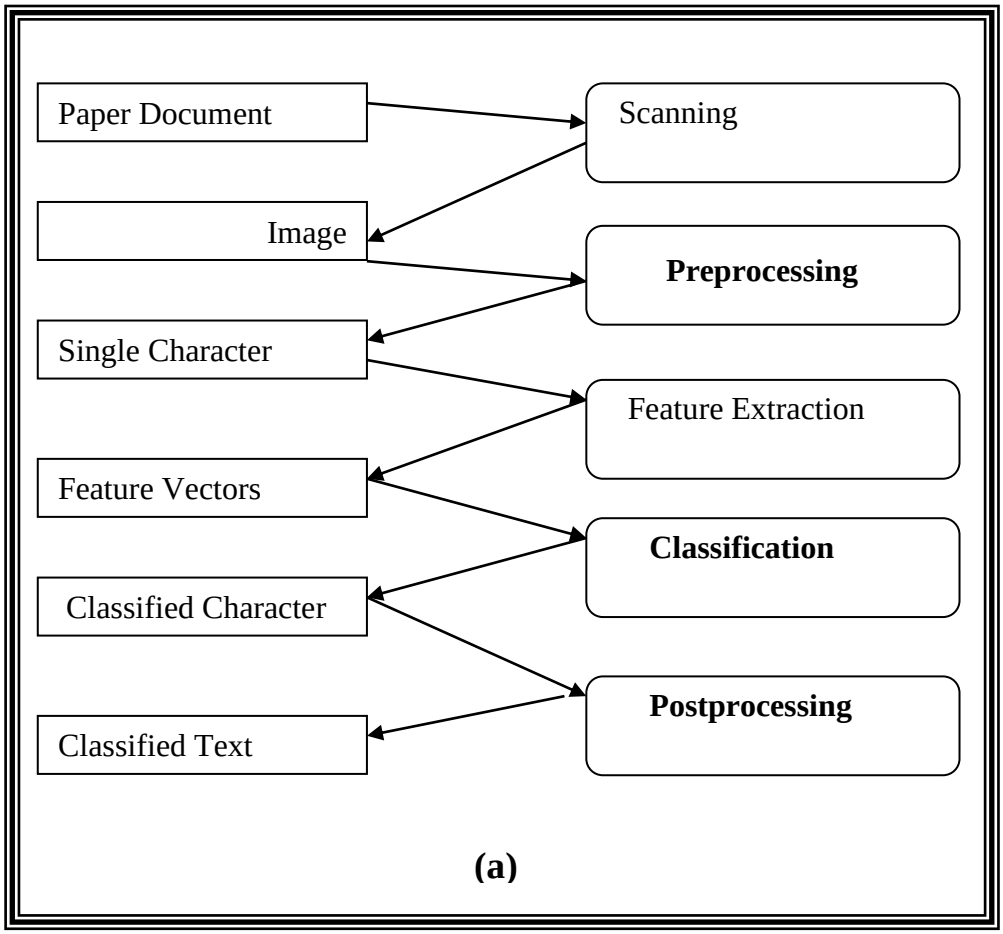
neural network) have a powerful function of pattern classification as a model for an artificial realization of human brain. We develop many algorithms for the improvement of recognized rate.

2- Optical Character Recognition (OCR)

Character Recognition or Optical Character Recognition (OCR) is the process of converting scanned images of machine printed (numerals, letters, and symbols), into a computer processable format (such as ASCII), [Amin, 2000, Bunke, 1997, Day,2000], (see figure, 1). This article describes the design of OCR systems and their applications.

3-Introduction to Machine Printed Arabic Character Recognition System (MACRS)

Machine Printed Arabic Character Recognition system (MACRS) aims to converting images document to text. The main objective of this section is to introduce a novel method of off-line machine printed Arabic character recognition used to construct *MACRS.*



(a)

Figure (1): Steps in a Character Recognition System

The following processing steps are taken when we design ***MACRS*** to carry out the task: scanning, thresholder, noise removal, perprocessor, feature extractor, recognizer (classifier) and post-processor.

3-1 Thresholder Operation of MACRS

Thresholding in **MACRS** is to extract a binary (0,1) image from the obtained digital image, which is then used for analysis in determining the value of the character. Image thresholding classifies the pixels of an image into the foreground (the writing) and the background. To convert the input image to binary and to extract the foreground from the background by thresholding we use a **histogram** of the pixel values in the image, there should be a large peak indicating the general value of the background pixels and another, smaller peak indicating the value of the foreground pixels.

Imthresh is a unit built in this project to extract a binary (0 - black and 1- white) image from the obtained digital image (Convert input image to binary image by thresholding).

$$\frac{\text{Syntax}}{\text{BW}} = \text{imthresh (Image name , level), converts the intensity image}$$

(image name) to black and white pixel store it into two dim. Array (BW)

3-2 Noise Removal Operation of MACRS

Thresholds on minimum component area and dimensions are used to discard small connected components corresponding to salt and pepper noise during the process. ***MACRS*** Perform two-dimensional adaptive noise-removal filtering by creating **AdaptF** unit.

The **AdaptF** unit applies a Wiener filter (a type of linear filter) to an image *adaptively*, tailoring itself to the local image variance. Where the variance is large, **AdaptF** performs little smoothing. Where the variance is small, **AdaptF** performs more smoothingly.

This approach often produces better results than linear filtering. The adaptive filter is more selective than a comparable linear filter, preserving edges and other high frequency parts of an image. In addition, there are no design tasks; the **AdaptF** unit handles all preliminary computations, and implements the filter for an input image. **AdaptF**, however, does require more computation time than linear filtering.

$$\frac{\text{Syntax}}{\text{J}} = \textbf{AdaptF (Image,[m n])}$$

Description

AdaptF filters an intensity image that has been degraded by constant power additive noise. **AdaptF** uses a pixel-wise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel. **AdaptF** filters the Image using pixel-wise adaptive Wiener filtering, using neighborhoods of size m-by-n to estimate the local image mean and standard deviation.

Algorithm

AdaptF estimates the local mean and variance around each pixel

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2) \qquad (1)$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \qquad (2)$$

where η is the *N*-by-*M* local neighborhood of each pixel in the image a. **AdaptF** then creates a pixel-wise Wiener filter using these estimates

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2}(a(n_1, n_2) - \mu) \qquad (3)$$

where ν^2 is the noise variance. If the noise variance is not given, **AdaptF** uses the average of all the local estimated variances.

3-3 Preprocessor Operation of MACRS

The preprocessor, as shown in figure (1), includes a segmenter and a normalizer. In ***MACRS***, the segmenter separates the bitmap array input into a plurality of smaller bitmaps. Normalizer includes three sub-modules, each of these three sub-modules performs one of three possible functions on each of the smaller bitmaps, and the function which each sub-module performs is distinct from the other two sub-modules. The three possible functions are thinning and thickening, size normalization, and slant correction.

3-3-1 Segmenter

The segmentation phase is a necessary step in ***MACRS***. Any error in segmenting the basic shape of machine printed Arabic characters will produce a different representation of the character component. In ***MACRS***, line separation is usually followed by a procedure that separates the line into words and after that to characters [straight segmentation]. In all printed

Arabic characters, the width at a connection point is much less than the width of the beginning character. This property is essential in applying the baseline segmentation technique, [Amin, 1997b, Amin, 1999, Amin, 2000], see equation 4.

$$V(j)=\sum W(i,j) \qquad (4)$$

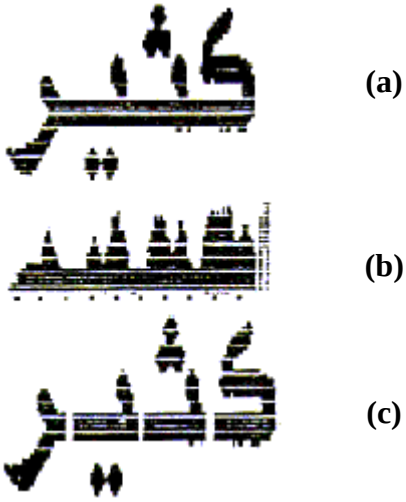
Where W(i,j) is either zero or one and i , j index the rows and columns, respectively, the connectivity point will have a sum less than the average value (AV).

$$AV = (I/Nc)\sum_{j=1}^{Nc} X_j \qquad (5)$$

And where Nc is the number of columns and Xj is the number of black pixels of the jth column, See figure(2).

3-3-2 Normalizer

A practical character recognizer must be able to maintain high performance regardless of the position, size and slant of a given character, so after the segmenter processes the bitmap array into a plurality of separate bitmap arrays, these separate bitmap arrays are fed to the normalizer for further processing.



(a)

(b)

(c)

Figure (2) An Example of Segmenter Process of Arabic Word into Characters (a) Arabic Word, (b) Histogram , (c) Word Segmented Characters.

The three sub-modules of the normalizer in ***MACRS*** will now be described. Each of these sub-modules acts to reduce the variance in the data ultimately fed to the recognizer system. The variance can be very large due to the variety of machine printed styles and the variety of documents on

which characters typically are printed. Both training and recognition efficiency of the recognizer can be increased by reducing this variance.

1-Thinning Module of MACRS

The problems outlined make the thinning process problematic if it is to be used as the first stage in a character recognition algorithm which is based on extracting features, we used in this research A Multi-Stage Process for Thinning Arabic Characters method to thinning machine printed Arabic character.

2- Size Normalization Module of MACRS

Size normalization for binary image f(x,y) applied in ***MACRS***, so that the size of the rectangle circumscribing the pattern is 16 x 16 pixels. Consequently the normalized image f'(x,y) is described as follows:

$$f'(x,y) = f(((width * x) / 16) + \delta x , ((height * y) / 16) + \delta y) \quad (6)$$

Where width and height are that of the pattern, respectively. Then δx and δy are the horizontal and vertical between the left-top corners of the image and the rectangle, respectively.

3- Slant normalization (Slant correction) Module of MACRS

The slant correction module in ***MACRS*** performs another variance-reducing operation on the bitmaps received from the segmenter before passing the bitmaps ultimately to the recognizer which is tolerant to slight slant and/or rotation. In general, the slant correction reduces slant and/or rotation by re-orienting the character represented by the bitmap array received from the segmenter to reduce, and preferably minimize, the overall width of the character.

In general, the process performed by the slant correction module corrects slant in characters by searching for a minimum width profile of a character using a binary search strategy on the angular slant of the character.

Slant correction in ***MACRS*** performs its slant correction function. The process implemented by the computer program utilizes the unit **Transform By Angle (X)** which performs transformations given by equation (7)

$$x' = x - (y * \tan(X)) \ , \quad y' = y \quad (7)$$

3-4 Character Feature Extraction Operation of MACRS

The key issue of ***MACRS*** is feature extraction, feature extraction stage in ***MACRS*** decomposes a normalized image of the character into numbers of features. This approach generally falls into global analysis technique using ***geometric moments invariants***. It receives binary image array (16 *

16 pixels) from normalizer creat feature extractor for it by calculating moment invariant.

A brief summary of the features and their sizes is given in table (1).

Geometric Moments Invariants	Value
First invariant moment	1.014080
Second invariant moment	0.471489
Third invariant moment	0.234804
Fourth invariant moment	0.020425
Fifth invariant moment	-0.001218
Sex invariant moment	-0.013615
Seventh invariant moment	0.000241

Table (1) Geometric Moment Invariant for Arabic Character (ع) Represent Input Data to Probabilistic Neural Network.

One of the fundamental issues in the design of an image recognition system is related to the selection of appropriate numerical features in order to achieve high recognition performance. Furthermore the geometric moment invariant used in *MACRS* as a feature extractor to extract an object invariant with respect to its position, size, and orientation. Moment provides characteristics of an object that uniquely represent its shape and, moreover, are invariant to linear transformations, [Yanjun, 1992].

In this study we will build a database for moment invariant (order 3) of 141 patterns (28 Arabic characters in four positions (isolated , at beginning , at end, at middle, (ي-ا ، ة ،لا،ء)) in the word , numeral character (9 - 0) and some special characters (+ , - , * , / , = , ! [,] , . , : , ؟ , ،)). The database as shown in table (2) had a total of 141 patterns in document (28 Arabic characters in four positions (isolated, at beginning, at end, at middle, (ي– ا ، ة ،لا،ء)), numeral characters (0–9) and special characters (+ , - , * , / , = , ! [,] , . , : , ؟ , ،) using simplified Arabic font and size 14, each character on the image document as a single bmp image file, preprocessing step in which original character image is transformed into a binary image by Thresholding, then noise removal is done, after that Arabic character is thinned size normalization and slant correction operation are done on each pattern. Final operation is to calculate invariant moment for each Arabic character (pattern) and store it into database (moment.txt file).

3-5 Neural Network Classifier Operation of MACRS

The advantage of using a neural network for Arabic character recognition is that it can construct nonlinear decision boundaries between the different classes in a non-parametric fashion, and thereby offers a practical method for solving highly complex pattern classification problems. Furthermore, the distributed representation of the input's features in the network provides an increased fault tolerance in recognition; thus character classification can occur successfully when part of the input is broken off and not present in the image, as well as when extra input signals are present as a result of noise. ,[Zurada, 1996]

	Isolated	End	Middle	Beginning
Alif	ا	ـا	ـا	ا
Ba	ب	ـب	ـبـ	بـ
Ta	ت	ـت	ـتـ	تـ
Tha	ث	ـث	ـثـ	ثـ
Jim	ج	ـج	ـجـ	جـ
Ha	ح	ـح	ـحـ	حـ
Kha	خ	ـخ	ـخـ	خـ
Dal	د	ـد	ـد	د
Dhal	ذ	ـذ	ـذ	ذ
Ra	ر	ـر	ـر	ر
Zan	ز	ـز	ـز	ز
Siin	س	ـس	ـسـ	سـ
Shiin	ش	ـش	ـشـ	شـ
Sadd	ص	ـص	ـصـ	صـ
Dad	ض	ـض	ـضـ	ضـ
Than	ط	ـط	ـطـ	طـ
Zah	ظ	ـظ	ـظـ	ظـ
Ayn	ع	ـع	ـعـ	عـ
Ghayn	غ	ـغ	ـغـ	غـ
Fa	ف	ـف	ـفـ	فـ
Qaf	ق	ـق	ـقـ	قـ
Kaf	ك	ـك	ـكـ	كـ
Lam	ل	ـل	ـلـ	لـ
Miim	م	ـم	ـمـ	مـ
Noon	ن	ـن	ـنـ	نـ
Ha	ه	ـه	ـهـ	هـ
Waw	و	ـو	ـو	و
Ya	ي	ـي	ـيـ	يـ

Lamalif	لا	ـلا	ـلا	لا
Tamabot	ة	ـة		
Hamza	ء			
Number	9 ، 8 ، 7 ، 6 ، 5 ، 4 ، 3 ، 2 ، 1 ،0			
Special char.	, + , - , * , / , = , !, [,] , . , : ؟ , ،			

Table (2) the database of MACARS

This is a very important characteristic for a recognition module in this application. We have chosen to implement a ***Probabilistic Neural Network (PNN)*** classifier. The PNN implementation attempts to model the actual probability distributions of classes with mixtures of Gaussians, allowing the computation of the posterior probability associated with each exemplar classification. PNN neural classification schemes that we used are essentially nearest-neighbor prototype matching. PNN algorithm adjusts the prototypes to approximate the density of exemplars in each class. If exemplars are uniformly distributed, the prototypes will uniformly fill the class boundaries. In our application, each class is represented by a single prototype. The resulting distribution of prototypes is such that they are approximately located at the mean of all exemplars, storage of all of the exemplars (moment database) in order to compute the final probability of class membership.

3-5-1 Probabilistic Neural Networks Classifier of MACRS

In ***MACRS*** the recognizer according to the invention receives the output of the feature extractor(moment invariant value) as its input. The recognizer module processes images to generate a "best guess" as to the identity of the character represented by the input bitmap and produces an output bitmap of that best-guess character. The recognizer is a probablistic neural network-based includes a fully-connected, three-layer neural network which accepts seven continious moment invariant values of image character. The disclosed embodiment of the neural network includes an input layer Radial basis layer, and an Competitive layer . The input layer includes 7 units, one for every seven moment invariant in an input bitmap. The competitive layer has 142 units whose activations vary from 1 to 0 . Each unit in the competitive (output) layer represents a different one of the 141 possible Arabic characters and last unit for rejection result. As a result of the recognition process, a bitmap of the characters corresponding to the output unit with the highest activation is produced as the output bitmap by the neural network-based recognizer.

Referring to table (1) moment invention of the charcter (ع), table (1), which was fed as input to the neural network produced the output

shown in table (3).The neural network output the correct character in response to the input . If "ع " was discovered to be the wrong character (e.g.,by a post processing procedure described below). In one embodiment, the output of the neural network-based recognizer is returned to the input of the pre-processor for additional processing by the system if the recognizer deems its output bitmap to be unacceptable (e.g., if the highest activation value of the output units is below predetermined threshold).

4- Experimental Results of MACRS

In order to measure the performance of ***MACRS*** on the character recognition problem, we describe in previous section and here three phases of analysis for using a neural network for machine printed Arabic character recognition: Data preprocessing and input data selection, neural network architecture and algorithm selection, and recognition results obtained a cross-validation study and noisy character data.

Output layer	Value	Calssifier value
(1,1)	1	0
(1,2)	2	0
(1,3)	3	0
(1,4)	4	0
(1,5)	5	0
(1,6)	6	0
.	.	0
.	.	0
.	.	0
(1,69)	69	1
.	.	0
.	.	0
.	.	0
(1,141)	141	0
(1,142)	142	0

Table (3) Output Layer in Probablistic Neural Network and the Result after Recognition of Arabic Character (ع).

4-1 Data Preprocessing and Feature Selection

The initial data sampled for the character recognition exercise consisted of a total 2961 patterns for a total 141 classes with roughly 21 pattern in each class , the (21) patterns are scanned and each pattern (image document) data was stored in one (.bmp or .jbeg) file. see figure (3). This contained information generated from the scan image document. Our

system needed persistent data storage. Each class represents a particular Arabic character with the 141 classes representing (ي-ا ، ة ،لاء) in four positions (isolated , at beginning , at end, at middle) in the word and for numeral characters (9-0) and special characters (+ , - , * , / , = , ! [,] , . , : , ؟ , ،). A total of 7 moments is extracted for character analysis as described in section (3-5). For each input vector **P = {X_1, X_2, …X_7},** its target output was represented as the vector **{t_1, t_2, … , t_{141}).** Here **Xi (1 ≤ I ≤ 7)** represents ***i***th window measurement and t_J **(1≤ j < 141)** is the target output for learning, i.e. if t_1 **= 1** and all other t_j are 0, then the input pattern is isolated "ا". Similarly, if t_2 = 1 and all other t_j are 0, then the input pattern represents an End "ـا". In this order, if the target output t_J = 1 in position j, then it represents a machine printed character of class j in sequence (ي–ة ، ا) , (9-0) , (+ , - , * , / , = , ! [,] , . , : , ؟ , ،) characters. The data finally presented to the neural network for training consisted of all 2961 input patterns with their respective target patterns. For all of our results, the data have been interleaved before presentation, This ensures better training since it allows the weights to adjust for the given problem in a well balanced manner, and for **Neural network development** we use recommended procedure Probabilistic Neural Network **(PNN)**, 7 x 7 x 142 architecture adequate for our purposes.

The algorithm uses parameter, learning rate η. This parameter allows the algorithm to converge more easily if it is properly set by the experimenter. In order to arrive at a reasonable values of this parameter for training we finally settle on a learning rate = 0.87.

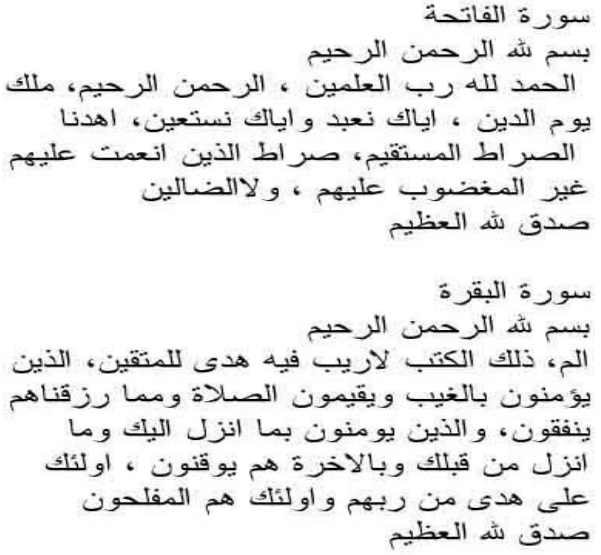
The networks were trained with sets until the accuracy rate on the training set was greater than 95%; the limit was set here in order to balance recognition rate while not tuning to finely the training set of data.

4-2 Recognition in MACRS with Cross-Validation

The neural network analysis of data is performed in two separate phases: training, when the network learns by example in an iterative manner, and testing, which then presents unseen data to be classified. In order to measure the performance of our neural network system on the character recognition problem, we use the cross-validation procedure.

Fu, [Amin, 1999b, Amin, 2000] describes the cross-validation process as: "K-fold cross-validation, repeats K times for a sample set randomly divided into k disjoint subsets, each time leaving one out for testing and the others for training". The value of K = 10 is usually recommended. Cross-validation requires that the original data set is split in k disjoint sets. At any one time, 90% of the data is used for training and the performance is tested

on the remaining 10%. Each training process is called a 'fold'. At the end of 10 folds, all data have been tested. In every fold, therefore, the training and test patterns are different. The overall performance of the system may be


سورة الفاتحة
بسم الله الرحمن الرحيم
الحمد لله رب العلمين ، الرحمن الرحيم، ملك
يوم الدين ، اياك نعبد واياك نستعين، اهدنا
الصراط المستقيم، صراط الذين انعمت عليهم
غير المغضوب عليهم ، ولاالضالين
صدق الله العظيم

سورة البقرة
بسم الله الرحمن الرحيم
الم، ذلك الكتب لاريب فيه هدى للمتقين، الذين
يؤمنون بالغيب ويقيمون الصلاة ومما رزقناهم
ينفقون، والذين يومنون بما انزل اليك وما
انزل من قبلك وبالاخرة هم يوقنون ، اولئك
على هدى من ربهم واولئك هم المفلحون
صدق الله العظيم

Figure (3) Machine Printed Arabic Image Document

سورة الفاتح~
بسم ل~ه الرحمن الرحيم
الحمد ل~ه رب العلمي~ ، الرحم~ الرحيم، ملك
يوم الدي~ ، اياك نعبد واياك نستعي~ ، اهدنا
الصراط المستقيم ، صراط الذين انعمت عل~~م
غير المغضوب عليهم ، ولا الضالين
~دق ال~ه العظي~

سورة البق~ة
بسم ل~ه الرحمن الرحيم
الم ، ذل~ الكتب ~~ريب فيه ~دى للمتقي~ ، الذي~
يومنون بالغيب ويقيمو~ الص~~ة ومما ررقناهم
ينفقو~ ، والذي~ يومنو~ بما انزل اليك وما
انزل من قبل~ وبالاخرة هم يوقنو~ ، اول~ك
على ~دى من ربهم واول~ك ~~ المفلحو~
~دق ال~ه العظي~

Figure (4) Text Document Result from MACRS for Machine Printed Arabic Image Document in Figure (3)

Measured using two different parameters: the average recognition data of training data in percentage (av. R_α), and the average recognition rate of the test data in percentage (av. R_β). As expected, the latter is smaller in practice but is very important since it represents the true performance of the neural network.

Table (4) shows the recognition performance using ten-fold cross-validation. Here, the recognition rate of the training and test data at the end of fold K ($1 \leq K \leq 10$) training is shown in separate rows of the table. The recognition rate in percentage represents the ratio of the total number of correctly classified patterns to the total patterns tested during a test phase. We follow rigid guidelines for specifying what is a correct classification. For a test pattern whose target is $\{t_1, t_2, \ldots, t_{141}\}$ and the actual output is $\{T_1, T_2, \ldots, T_{141}\}$, the correctly classified pattern must satisfy the condition $T_j - t_J < 0.2$ for all j ($1<j<141$). If this condition is violated even once in a pattern, then it is misclassified. Similar stringent guidelines are followed for training. The training process for the network is stopped only when the sum of squared error falls below 0.0001.

It is important to note here that the system performs extremely well with recognition rates ranging between 84% and 88% on different folds and the overall recognition is 85.8%. This is a very good performance taking into account the fact that we have a limited number of samples in each class and that a linear discriminant analysis yields a recognition rate of 56% at best. The recognition on the training data is also extremely high, 99.8%, which represents a very good training.

Fold	**Recognition Rate % Training (R_α)**	**Recognition Rate % Testing (R_β)**
1	99.9	85.0
2*	99.9	87.0
3	99.7	87.0
4	99.8	84.0
5	99.8	86.0
6	99.9	86.0
7	99.9	85.0
8	99.9	85.0
9	99.9	88.0
10	99.8	85.0
Average	99.8	85.0

Table (4) Neural Network Recognition Rate Performance Using Ten Fold Cross-Validation. Recognition Rates on the Training and Test Sets.

In Table (4), the results have been produced keeping the experimenter bias to a minimum when developing a neural network for analysis and the feature extraction stage. These sets of results, however, do not tell us about the quality of our feature extraction in terms of their resistance to noise. In other words, we need to quantify how well the system will perform in the presence of noise. For this purpose we generate Gaussian noise with a fixed distribution (mean 0, sd. = 1) and use this to contaminate our character recognition data. Recognition rates are then recorded for varying noise amplitude. For further experimentation we do not follow cross-validation since our aim is not to investigate the true generalization error, rather it is to quantify the degradation in performance with pre-defined step-wise increases in noise. For this purpose, data in fold 2 , Table (4) is selected (marked with an asterisk). We train with 90% of the data in the training set and test with 10% of the data in the test set, when injected with additive non-cumulative noise of varying amplitude. The noise data is generated using a Matlab function library (imnoise (I,type). The noise vector N is a series of randomly generated numbers which is transformed within the [-1, +1] range. A total of ten trials is conducted, each time varying the maximum offset allowed. The maximum noise offset δ represents the maximum noise possible for a single pattern. The actual noise value for a particular pattern with the [-1, +1] range is multiplied by this maximum offset before being added to the character data. The average noise $\check{N}$ represents the ratio of the total noise present in the data and the number of patterns. This value for a particular trial is always much below the noise offset δ for that trial. Since noise is random, the average noise $\check{N}_{\alpha}$ for training data is different to the test data $\check{N}_{\beta}$. For different trials, we use different noise series but with the same noise distribution. During each trial, the neural network is trained and tested with noisy character data. Table (5) shows the recognition results obtained using the above procedure.

In table (5), the average noise per pattern added to both the training and test set is shown with the recognition rates obtained on both the training and test set when the neural network learning was finished. As expected, in every successive trial, the amount of noise added to the system increases. The training recognition rates fall and then start to rise: this phenomenon has been noted in other studies when the presence of noise actually helps the neural network for training purposes. Following this trend, the test performance also degrades with increasing noise for most cases. Some important points of observation may be stated as follows:

· The degradation in performance is graceful and predictable. The correlation r between the amount of noise and the drop in recognition rate is high, r_{train}= -.85, and r_{test} = -.86.
· The recognition rates are high for most trials except when the noise increases considerably.
· The degradation in training and test recognition rates are highly correlated r = .97 but in most cases the degradation in performance is not directly proportional.

Trial	Noise	Av. α Noise Ň α	Recognition Rate % Training (R_α)	Av. β Noise Ňβ	Recognition Rate % Testing (R_β)
1	0.1	0.02	99.9	0.03	85.0
2	0.2	0.06	96.7	0.07	81.0
3	0.3	0.07	93.0	0.08	74.0
4	0.4	0.10	90.5	0.12	75.0
5	0.5	0.128	87.6	0.13	71.0
6	1.0	0.28	73.7	0.37	58.0
7	1.5	0.40	76.2	0.45	56.0
8	2.0	0.51	66.8	0.61	51.0
9	2.5	0.59	68.2	0.90	48.0
10	3.0	0.70	77.7	1.23	52.0

Table (5) Neural Network Recognition Rate Performance with Noisy Machine Printed Character Data. Results are Shown for Gaussian Additive Noise Added to both the Training and the Test Set.

4-3 Compare MACRS with Previous Systems

To evaluate the performance of ***MACRS*** , we compare the recognition rate of ***MACRS*** with other OCR software's by scan image document as shown in figure (3) and execute this image document on ***MACRS, ALKARI AL-ALE*** to its recognized characters, (see figure 4 and 5). We compare the results of recognized image document from three software's, we find that ***MACRS*** has recognized rate 87% better than ***ALKARI AL-ALE*** where ***ALKARI AL-ALE*** has 84% recognition rate.

Table (6) summarized previous approaches in Arabic Machine printed characters recognition. However, to give a estimate of relative performance, we have included this table for completeness.

Recognition Rate	MACRS	ALKARI AL-ALE
	87%	84%

Table (6) Summary of Previous Approaches in Handwritten Arabic Characters with MACRS

5 - Conclusion

MACRS is an optical machine printed Arabic character recognition (OCR) software capable of producing a fully-editable electronic document with accurate character recognition (85%) for machine printed character recognition.

MACRS scans any machine printed document. Scan directly into your favorite Windows applications word processors format to correct OCR results before converting to another application. This means that you can edit machine printed text without retyping quickly, easily, and above all, accurately. Actual results may vary depending upon the document.

سورة الفاتح~
بسم ل~ه الرح~ن الرحي~
الحمد ل~ه رب العلمي~ ، الرحم~ الرحي~، ملك
يوم الدي~ ، اياك نعبد واياك نستعي~ ، اهدنا
الصراط المستقي~ ، صرا~ الذين انعمت عل~~م
غير المغضوب عليهم ، ولا ال~الي~
صدق ال~ه العظي~

سورة البق~ة
بسم ل~ه الرح~ن الرحي~
الم ، ذل~ الكتب ~~ريب فيه ~دى للمتقي~ ، الذي~
يومنون بالغيب ويقيمو~ الص~~ة ومما ررقنا~~
ينفقو~ ، والذي~ يومنو~ بما انزل اليك وما
انزل من قبل~ وبالاخرة ه~ يوقنو~ ، اول~ك
على ~دى من ربهم واول~ك هم المفلحو~
صدق ال~ه العظي~

Figure (5) Text Document Result from *ALKARI AL-ALE* for Machine Printed Arabic Image Document in Figure (3)

6-1Summary of Contributions

After running ***MACRS*** software and experimentation on samples for machine printed. Arabic (machine printed characters compliant data set of 2961 characters) we find the following assure points .

- moment-invariant features for machine printed characters tuned to produce relevant features for Arabic recognition from data coordinates while reducing the input space.
- Probabilistic neural network tuned to recognize the 141 character classes in an easy and powerful recognized way.
- Accurate recognition rate for ***MACRS*** is 99% for training data set and 85% for test data set. The accurate recognition rate for ***MACRS*** is a good rate and best results when we compare theses rates with the accurate recognition rate of previous researches and software's of OCR.

6-2 Conclusion

Upgrading to ***MACRS*** gives you access to a powerful new user interface including accuracy increasing features such as advanced zone editing, proofreading, and saving training data. That's because ***MACRS*** OCR is the best in its class. Take a look at some of these features: **Improved Character Accuracy Up to 80%* more accurate :** Easily turn any machine printed document image into electronic documents without retyping. Dramatically saves time. **Full Document Recognition** : Accurately recognizes even the most complex of document. **Proofreading** : Proofread and edit documents directly from within ***MACRS*** for even more accurate results. During the proofreading process, ***MACRS*** provides you an image window to view the original document and accept or correct any word that ***MACRS*** suspects may not be recognized accurately. **Page Type Templates** : Optimizes OCR results by document type (for example, letter, magazine article, and so on).

REFERENCES

[1] Abuhaiba, I.S.I. and Mahmoud, S.A. (1994). Recognition of Handwritten Cursive Arabic Characters, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No 6 : 664-672.

[2] Amin, A. (1997a).Off-Line Arabic Character Recognition : The State of Art, Pattern Recognition, Vol. 31, No. 5 :517:530.

[3] Amin, A. and Motawa, D. (1997b). Segmentation of Arabic Cursive Script, 4th International Conference Document Analysis and Recognition (ICDAR 97).

[4] Amin, A. (1998). Off-line Arabic Character Recognition - the State of the Art [review], Pattern Recognition, Vol. 31, No. 5 : 517-530.

[5] Amin, A. and Mandana, K. (1999) Automatic Recognition of Printed Arabic Text Using Neural Network Classifier, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 1.

[6] Amin, A.(2000). Recognition of Printed Arabic Text Based on Global Features and Decision Tree Learning Techniques, Pattern Recognition, Vol (33), No (8), :1309-1323.

[7] Amondon. R, and Srihari, S.N. (2000). On-line and off-line Handwriting Recognition: A Comprehensive Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1 : 63-84.

[8] Bourbakis, N. and Gumahad, A. (1991). Knowledge-Based Recognition of Typed Text Characters, Character & Handwriting Recognition: Expanding Frontiers, World Scientific Publishing Co., Singapore.

[9] Bunke, H. and Patrick, S.P. (1997). Handbook of Character Recognition and Document Image Analysis. World Publishing Scientific, Singapore.

[10] Day, S.P. (2000). The Way to Program a Neural OCR, Electronic Engineering Times, Issue 845.

[11] Klassen, T. (2001). Towards Neural Network Recognition of Handwritten Arabic Letters. Masters Thesis, Dalhousie University, Halifax, U.S.A.

[12] Yanjun, L. (1992). Reforming the theory of invariant moments for pattern recognition. Pattern Recognition, Vol. 25, No. 7 : 723-730.

[13] Zurada, J. M. (1996). Introduction to Artificial Neural Systems, West Publishing Co.